**Project: Doch**

My project, Doch, which means embroidery in Balochi language, is a mobile application that can be deployed as an android app and as an iPhone app. Doch is a marketplace app yet being developed where users can find fine Balochi embroidery by scrolling through an image feed just like some social media applications such as Instagram. Images in the image feed will be uploaded by the users. App will allow the users to favorite any picture they like, they will be allowed to comment, and they will also be allowed to contact other users through direct messages. Users will receive notifications based on any action performed to either shared pictures such as comments and favorites or if they receive a direct message.

My motivation behind developing this app is that I feel like Balochi embroidery is yet so beautiful and unique and still it is so unpopular in the World or even in Pakistan. Secondly, I want this app to become a bridge between the artist who can make these beautiful embroideries and the customers who want to buy Balochi embroidery. I just want this app to become a platform for these brilliantly skilled women who have mastered the art of Balochi embroidery get recognition and get paid the money they deserve. I also want this app to become a platform for the interested buyers, who are fascinated by Balochi embroidery clothes but do not know where they can buy these from. Balochi embroidery clothes are not sold in shops or markets and that is why a lot of people cannot buy Balochi embroidery clothes for themselves. My app’s mission is to make Balochi embroidery available for anyone who wants to buy it.

To develop this mobile application, I am using python programming language along with kivy and kivymd. Kivy is a cross-platform python framework used to develop applications for android, iPhone, and web applications. Kivymd is basically a collection of material design compliant widgets that I use with kivy to build the application. Kivy language code can be implemented in two ways either you create a file with extension. kv or you write the kivy code inside a multiline comment inside the .py file for your projrct. I have chosen the later one, because I was getting errors when I tried to create a scrollable image list which represents my apps image feed, but with the kivy code in the .py file of the project it resolved my problem.

In my kivy code, first I have declared a ScreenManager which manages the different screens I have in my application such as WelcomeScreen, and LoginScreen. Then all of the screens that my app contains are declared inside the ScreenManager. Once the screens are declared then they are designed using various kivy and kivyMD widgets. These screens have text labels that shows a text on that screen, buttons that perform a certain job when pressed, text fields to get information from the users and pop-up dialogue boxes that give errors when something goes wrong. The mainscreen has more attributes than the other screens. It has a MDBottomNavigation which has four navigation buttons in the bottom and each of the button switches to a different screen.

In the python code, I have created classes for each screen that take the argument ‘Screen’ and also a ScreenManager object is created to add all of the screens to it as widgets. Then DochApp class is declared which is the driver of the application. Inside DochApp class, I have declared functions for signup, and login in which the logical part of those screens go into. I use a real time database of firebase in .json format to save the signup information of new users and then later check if the credentials are correct when a user attempts to login. I have also put some conditional logic for my login and signup screens which basically check if email, username or password is a one word and check if user leaves a text filed empty. In the login function, I request the information from the firebase using a authentication key, and the program also verifies if the password matches a particular registered email.